# Earth System Science Server (ES3): Local Infrastructure for Earth Science Product Management

James Frew
Donald Bren School of Environmental Science and Management
and
Institute for Computational Earth System Science
University of California
Santa Barbara, CA 93106-5131

***Abstract* – Earth System Science Server (ES3) is a suite of technologies for managing end-to-end science data product creation, maintenance, updating, and dissemination. ES3 has four main components: *Metadata management* keeps track of external data objects imported, processes run, data objects created, data products delivered, and lineage (ancestor-descendent relationships between processes and data streams. *Process management* uses a generic Linux cluster to perform both science and production processing. *Data management* keeps all data online in mirrored server pairs, acquired "just-in-time" to leverage plummeting hardware costs. *Data product dissemination* exposes products via three mechanisms: well-know URIs that follow community-standard naming schemes and support OpenDAP services; seamless layers in a pre-computed hierarchy of resolutions, via the Microsoft TerraServer; and library-style searches using the Alexandria Digital Library web services. The ES3 prototype will manage the creation, updating and distribution of snow-covered area, snow albedo, and snow water equivalence data products.**

## I. Introduction

Current computing environments and research practices provide little support for bridging the gap between research and production. A researcher responsible for creating data products won't necessarily record the details of inputs, outputs, and processing steps, so that publishing these products with adequate metadata usually requires reorganizing the data and creating or rediscovering metadata values. Dissemination, especially where custom processing such as subsetting, re-projection, or reformatting is required, is often treated in a similarly *ad hoc* fashion.

Earth System Science Server (ES3) is an umbrella term for a local infrastructure for Earth science product management. By "local," we mean the infrastructure that a scientist uses to manage the creation and dissemination of her own data products, particularly those that are constantly incorporating corrections or improvements based on the scientist's own research. A local infrastructure, in addition to being robust and capacious enough to support public access, must be flexible enough to manage the idiosyncratic computing ensembles that typify scientific research.

ES3 is a suite of technologies for managing end-to-end science data product creation, maintenance, updating, and dissemination. ES3 has four main components: metadata management, process management, data management, and data product dissemination.

*Metadata management* keeps track of everything that happens in the system -- external data objects imported, processes run, data objects created, and data products delivered. ES3 also captures and maintains lineage metadata, i.e. the ancestor-descendent relationships between processes and data streams that are absolutely critical for establishing the pedigree and credibility of investigator-generated data products.

*Process management* uses a generic Linux cluster to perform both science (e.g. algorithm development) and production (e.g. custom subsetting) processing.

*Data management* keeps all data online in RAID arrays with 100% off-site mirroring. ES3-managed online objects are accessible via their object identifiers; a database maintains the correspondence between object identifiers and file names or database queries. Storage is managed as mirrored server pairs, acquired "just-in-time" to leverage plummeting hardware costs.

*Data product dissemination* exposes products via three mechanisms:

- well-know URIs that follow community-standard naming schemes and support OpenDAP services;
- seamless layers in a pre-computed hierarchy of resolutions, via the Microsoft TerraServer; and
- library-style searches using the Alexandria Digital Library web services.

## II. Metadata Management

The metadata management component is the heart of ES3, keeping track of everything that happens in the system – external data objects imported, processes run, data objects created, and data products delivered. Metadata about all these activities are captured by the "Lab Notebook" (LN) subsystem [3] (Figure 1). Scripting language (Perl, Python, sh, etc.)

This work was supported by the U.S. National Aeronautics and Space Administration under Cooperative Agreement NNG04GC52A.

"wrappers" use the LN network application programmer interface (API) to create and send XML metadata about ES3 processes or events to the LN database.

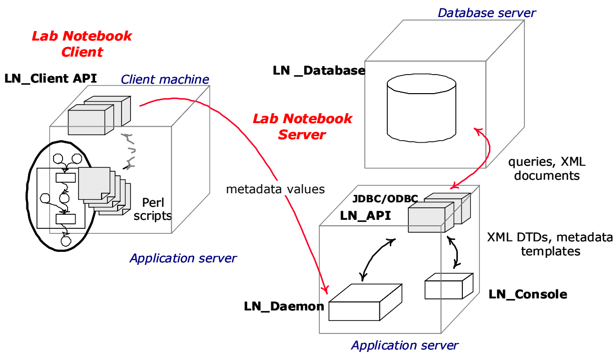


Figure 1. Lab Notebook subsystem.

Wrappers can be either synchronous (invoking, or invoked by, their corresponding process or event) or asynchronous (invoked independently to simulate or record a process or event.) A unique feature of the Lab Notebook is its ability to capture and maintain lineage metadata [2], i.e. ancestor-descendent relationships between processes and data streams that are absolutely critical for establishing the pedigree and credibility of investigator-generated data products.

Metadata are accessed within ES3 by querying the LN database through a variety of special-purpose clients. In particular, clients are provided to administer the system, to maintain a system-wide metadata registry, and to create the metadata infrastructure (XML DTDs, database tables, etc.) that supports new science data products. New clients can be built using the LN network API (currently bound to Java and Perl) or the LN database's own client access mechanisms (e.g., JDBC). The LN network API includes special methods that dynamically reconstruct either forward (descendent) or backward (ancestor) lineage graphs (Figure 2) for any process or object known to the LN.

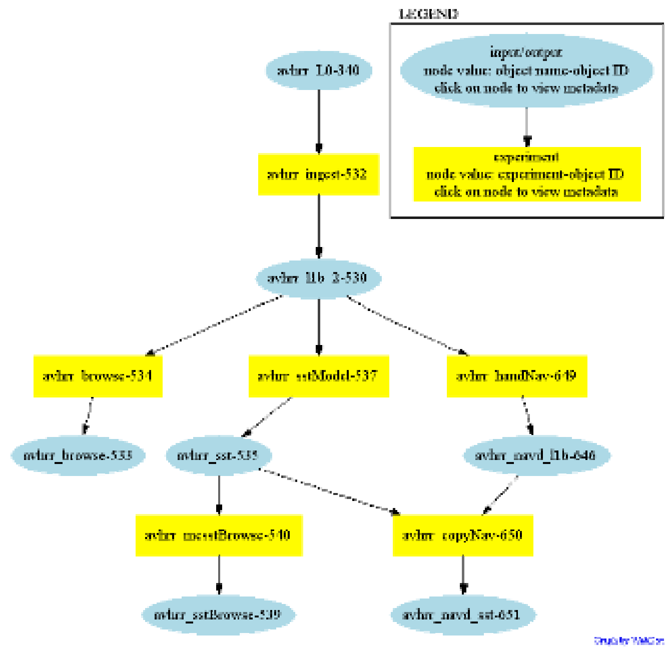


Figure 2. Eaxmple lineage graph.

## III. PROCESS MANAGEMENT

The process management component of ES3 uses the LN database to assemble the processing environment (commands, parameters, and input data streams) necessary to create (or recreate) any existing or potential object. This is predicated on using synchronous wrappers, and saving stable references to all necessary processed and wrappers in the LN. In almost all cases, the processing environment of choice is a commodity cluster, since it offers the best performance per unit cost currently available, and most of our processing algorithms are parallel. ES3 uses the NPACI Rocks software suite [5] to manage clusters of Linux processors. Processing environments (re-)created by the LN are submitted to the Rocks scheduler as batch jobs. In the ESSW ESIP, we have found that custom satellite product generation in this environment typically completes in a few seconds. We have also found it trivial to add additional computing capacity to the cluster. New or even temporarily unused machines can be "inducted" into the cluster in less than 20 minutes, so we can use the computing power of hundreds of machines if necessary.

## IV. DATA MANAGEMENT

The data management component of ES3 keeps all data online. Given the plummeting cost of online storage -- 50% per year -- complex storage schemes like disk-tape hierarchies cannot be justified for the storage volumes and support levels at which ES3 is targeted. Both file and database storage are supported. ES3-managed objects are accessible via their object identifiers; a database maintains the correspondence between object identifiers and file names or database queries.

Apart from object identifier management, the ES3 data management component is completely generic. A typical ES3 installation will have a few commodity file servers supporting several terabytes of RAID storage, accessible via NFS. Since ES3 makes no further assumptions about storage configuration, both servers and storage can be added "just in time" to take maximum advantage of declining hardware costs (and increasing storage density -- available physical space being an increasingly important constraint on system implementation.) Our current strategy is to purchase storage in units of fully redundant servers (one primary and one backup server) (Figure 3); as of May 2004 this costs less than US$3000 per (redundant) terabyte served.

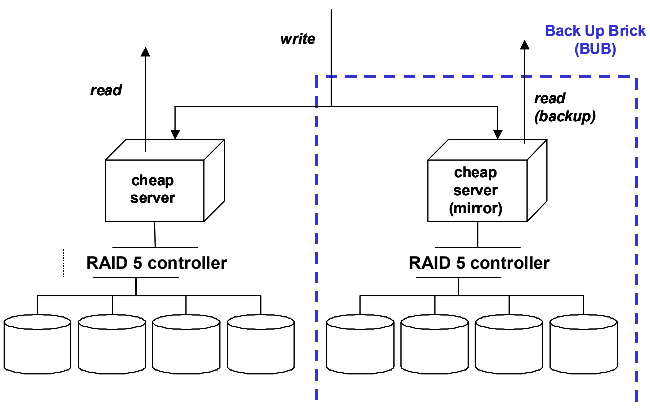


Figure 3. Online storage configuration.

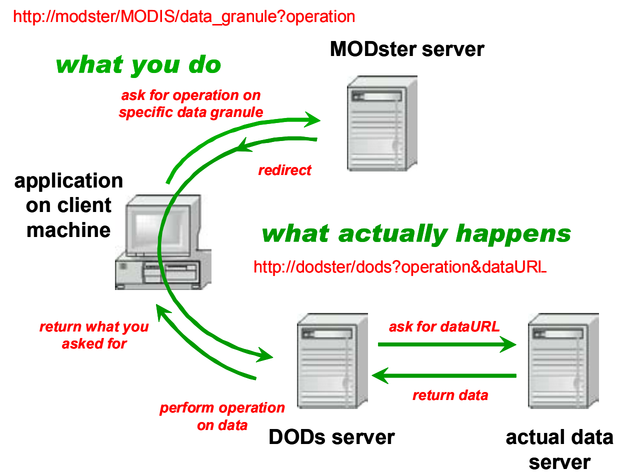


Figure 4. Third-party processing using HTTP redirects.

## V. DATA PRODUCT DISSEMINATION FOR DIFFERENT USER GROUPS

The data product dissemination component of ES3 is designed to make ES3-managed products accessible to as large and diverse a community as possible:

*Information providers* are assumed to already know what products they want; they will be the least tolerant of barriers to immediate delivery of those products. These users are offered direct access to standard or custom products via Web services. Since ES3 keeps everything online, it is trivial to make an ES3 data object accessible via a URL. Where appropriate, access to standard transformations (e.g. thresholding, subsetting, etc.) of an object will also be provided using OpenDAP, either directly with an ES3 OpenDAP server, or indirectly with a third-party Open-DAP server that has announced its availability through the DODSter redirection service (Figure 4).

*Information browsers* are assumed to be reasonably familiar with a data product domain, but not necessarily with the scope or character of an ES3's holdings in that domain. They may also wish to perform exploratory analyses on the domain to help identify products subsets of interest. For these users, ES3 will support random access to selected data products through a Microsoft TerraServer interface [1]. Products to be made available through this interface are hierarchically decomposed into nested tiles, which are stored in the TerraServer database along with metadata to support their rapid retrieval. In addition to supporting near-real-time "pan-and-zoom" visualization of the product, TerraServer queries can also be used to rapidly identify and examine arbitrary subsets of interest.

*Information seekers* are assumed to have a constrained notion (e.g. geophysical parameter, region, season, etc.) of what they seek, but may be unfamiliar with the corresponding providers and products. In addition to supporting the standard Federation FIND interfaces, ES3 includes the Alexandria Digital Earth Prototype (ADEPT) digital library middleware [4], which allows an ES3 system to function as a node in the Alexandria distributed geographic library [6]. Specifically, the ADEPT middleware exposes standard Web services for collection description, querying, and metadata report retrieval. To an Alexandria user, an ES3 system appears to be a library; products appear to be a collection; and an individual object or service within a product appears to be a holding within a collection (Figure 5). In addition to standard library metadata, ADEPT metadata reports allow access to browse imagery and product content.

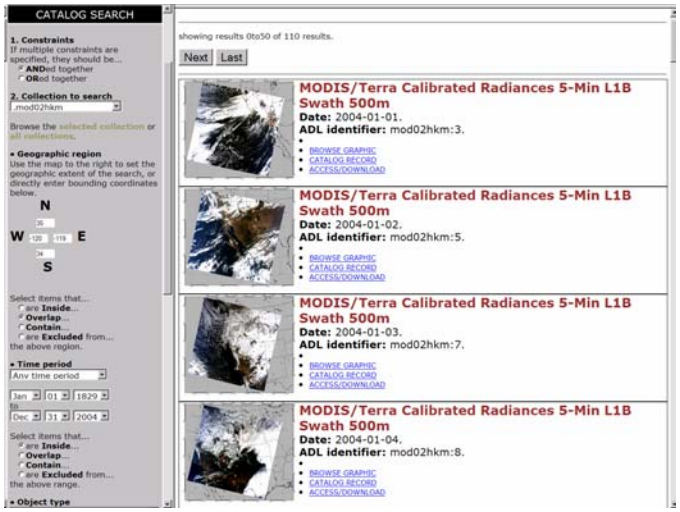


Figure 5. ADEPT library interface to MODIS collection.

## VI. THE SNOW SERVER

The prototype application of our ES3 technologies will be a system to manage the creation, updating and distribution of our three snow products -- snow-covered area, snow albedo, and snow water equivalence. In all cases, the overriding goals of the ES3 are to provide easy access to customized subsets of trustworthy products.

Ease of access is supported by ES3's multiple modes of making products available: they can be discovered via the Alexandria digital library, browsed and analyzed via the TerraServer, and retrieved via Web services (simple GETs, OpenDAP, or the TerraServer's .NET interfaces). Customized subsets are supported by allowing all products to be subset, aggregated, or composed by space, time, and/or parameter. The ES3 will also maintain user profile information, both to simplify customizations for repeat users and to allow the system to be tuned to better support frequent requests (e.g., by pre-computing popular subset combinations). Subsets will be implemented either in response to OpenDAP or .NET requests, or in real-time where supported by the TerraServer's ES3-extended capabilities.

Trustworthy products are supported by using the Lab Notebook's lineage management capabilities to rigorously track the complete processing history of all products. Users of snow ES3 products will see this both in the lineage information they receive with their products (or request later), and in the alerts they will optionally receive when subsequent reprocessing invalidates or otherwise affects their products. Put simply, it will always be possible to tell exactly where a snow property dataset came from and how it was prepared, and it will always be possible to determine which products were derived from any ES3 object or process.

## VII. CONCLUSION AND FUTURE DIRECTIONS

An ES3 system is an installation of ES3 components, under common administration, functioning either parallel to, or as an integral part of, an Earth scientist's computing environment. When deployed alongside an existing computing environment, an system's primary role is to augment that environment with capabilities and services that make existing scientific information more widely and reliably accessible. This goal of "non-disruption" of current environments is carried over from ESSW, and distinguishes ES3 from information management approaches that require everything to be under the control of a single system (DBMS; GIS; etc.). This aspect of ES3 should make it especially attractive to investigators that already have a significant investment in a scientific computing infrastructure (hardware, software, and modes of use) that they do not wish to abandon. For these users, wrapper scripts and networked file systems form the primary points of tangency between ES3 and their existing environments. ES3 handles the "public" aspects of their products, and their current environments continue to support research and development.

We also expect that ES3 will serve as a nucleus for establishing new scientific computing environments, since its key components cover many of the capabilities these environments will require. For sites that do not have already have significant investments in processors or storage, the ES3 approach will offer much higher performance/price ratios than single-vendor solutions. ES3's Lab Notebook approach to metadata acquisition and management may be attractive to new sites by not foreclosing future options; whereas if a new site chooses to manage all its metadata in (say) a GIS, then that becomes a commitment to that particular technology. The current best practices in Earth system science product management are ad hoc; there is no turnkey "science data system in a box." ES3 moves towards that goal by establishing simple interfaces to product management capabilities, implementing them with state-of-the-art components, and operating them with minimal impact on a site's research activities. ES3 bridges the gap between research and production, as painlessly as possible for the scientists responsible for both.

## ACKNOWLEDGMENT

The team designing and building ES3 includes Rajendra Bose, Michael Colee, Deborah Donahue, and Peter Slaughter.

## REFERENCES

[1] T. Barclay, J. Gray, and D. Slutz, "Microsoft TerraServer: a spatial data warehouse," Microsoft Corporation, Redmond, WA Technical Report MS-TR-99-29, 2000.

[2] R. Bose, "A conceptual framework for composing and managing scientific data lineage," in *SSDBM 2002 - 14th International Conference on Scientific and Statistical Database Management*, Edinburgh, 2002, pp. 15-19.

[3] J. Frew and R. Bose, "Earth System Science Workbench: a data management infrastructure for Earth science products." in *SSDBM 2001 – 13th International Conference on Scientific and Statistical Database Management*, Fairfax, VA, 2001, pp. 180-189.

[4] G. Janée and J. Frew, "The ADEPT digital library architecture," in *Second ACM/IEEE-CS Joint Conference on Digital Libraries*, Portland, OR, 2002, pp. 342-350.

[5] P. M. Papadopoulos, M. J. Katz, and G. Bruno, "NPACI Rocks: tools and techniques for easily deploying manageable Linux clusters," in *CLUSTER'01 - 3rd IEEE International Conference on Cluster Computing*, 2001, pp. 258-270.

[6] T. Smith, J. Frew, G. Janée, and L. Hill, "The Alexandria Digital Library Project," in *NIT2001 12th International Conference on New Information Technology - Global Digital Library Development in the New Millenium*, Beijing, 2001.